

I hereby certify that this paper is being deposited with the United States Postal Service as Express Mail in an envelope addressed to: BOX PATENT APPLICATION, ASSISTANT COMMISSIONER FOR PATENTS, Washington, D.C. 20231, on this date.

January 11, 2002

Date

Express Mail No. **EL846174092US**

Dale Auman

1 Inventors: Eric DeLano

2 CUSTOMIZED PORTS IN A CROSSBAR AND
3 METHOD FOR TRANSMITTING DATA BETWEEN
4 CUSTOMIZED PORTS AND SYSTEM AGENTS

5 [0001] The present invention generally relates to customized ports
6 in a crossbar. More specifically, it relates to customized ports in a crossbar
7 and a method for transmitting data on the customized ports in a crossbar.

8 [0002] A crossbar is a chip or chip component commonly used to
9 provide a processing system of high frequency links between multiple ports
10 and system agents. Generally, each port is designed as an input port for
11 receiving data from system agents (i.e., source agents) and an output port
12 for transmitting data from the input ports to other system agents (i.e.,
13 destination agents). The system agents can include multiple computer
14 components, such as a central processing unit (CPU), Input/Output (I/O)
15 controllers, memory controllers and cache memory.

1 [0003] More particularly, the requesting agents are generally the
2 CPUs, which can demand different amounts of data depending on the
3 particular requests. The responding agents are generally the Input/Output
4 (I/O) controllers, memory controllers and cache memory, which are also
5 defined by various configurations and bandwidth requirements. As a result
6 of the various agents and the multiple ports, a crossbar is used to sort out
7 the high-frequency links to meet the demands of these competing system
8 agents for multiple ports, which may comprise thousands of different ports.
9 To further the complexity of the processing system, each of the ports are
10 generally divided into multiple virtual communication channels in order to
11 fully utilize the potential of each port.

12 [0004] The ports in a crossbar typically have the same fixed
13 bandwidth for transmission. The problem with a typical crossbar
14 configuration is that it does not effectively deal with the demands of
15 different bandwidth needs for the requesting and responding agents. For
16 example, a requesting agent may require only a fraction of the fixed
17 bandwidth for a particular transmission. Nevertheless, the full fixed
18 bandwidth is used, since all ports in a typical crossbar have the same fixed
19 bandwidth. As a result, bandwidth is wasted in the process. On the other
20 hand, if there is not enough fixed bandwidth designed into the crossbar,
21 performance will also suffer. Furthermore, multiple processors have been
22 used within a single computer, which requires more flexibility in dealing
23 with the different available bandwidth needs. Thus, there is a need for an
24 improved crossbar and method for transmitting data between various
25 system agents and ports.

BRIEF SUMMARY OF THE INVENTION

[0005] The present invention is directed to customized ports in a crossbar. More specifically, it relates to customized ports in a crossbar and a method for transmitting data on the customized ports in a crossbar.

[0006] The present invention provides a crossbar for providing connections between a plurality of ports and a plurality of system agents via a processing system, which includes a plurality of ports, with each port being capable of being an input port customized for receiving data from a source agent and an output port customized for transferring data to a destination agent, and crossbar control data for specifying crossbar control information for transferring data from an input port to an output port having different port configurations.

[0007] The present invention also provides a method for transmitting data between customized ports and a plurality of system agents in a processing system via a crossbar, which includes the steps of receiving data on an input port, obtaining the destination output port for the data received on the input port, determining whether the input port has the same configurations as the output port, obtaining control information from the crossbar control data when the input port does not have the same configurations as the output port, processing the data according to the obtained control information from the crossbar control data, and transmitting the processed data to the destination output port.

DESCRIPTION OF THE DRAWINGS

[0008] FIGURE 1 is a schematic diagram of a simplified prior art crossbar;

1

3

5

7

9

10

23

1 Each port is typically designed as an input port for receiving data from a
2 source agent and an output port for transmitting data from the input port to a
3 destination agent. In particular, as shown in FIG. 1, port0 12 is designed for
4 an input port 20 with an 8 bit data width and an output port 22 with also an
5 8 bit data width. In other words, an 8 bit length data width is allowed for
6 transmission on port0 12 as input data or output data. Generally, if a
7 connection via the crossbar is N bit wide and operates at a bandwidth of X
8 bit per second, M bit of data will complete a transmission in $M/(N \cdot X)$
9 seconds. Data received on the input ports typically includes control data 24,
10 which indicates information and/or destination output port relating to the
11 data.

12 [0015] The crossbar, in turn, uses the control information from
13 the control data 24 to negotiate a connection with one of the ports. The
14 crossbar typically uses the control information to verify the validity of port
15 information relating to the data and to obtain the requested destination
16 output port. In this case, a 1 bit length to validate the request, and a 2 bit
17 length to specify the destination port. Control data is well-known in the art,
18 and they vary in bit length and are generally sent from the source agent
19 along with the data received on the input port. A typical crossbar is
20 designed to have the same bandwidth for the input and the output of each
21 port. Thus, every port in the crossbar typically has the same bit length
22 implementation.

23 [0016] In the exemplary prior art crossbar shown in FIG. 1, port1
24 14 is similarly designed as an 8 bit input port 26 and output port 28 with a 3
25 bit length for control data 30 to validate the input data and provide a
26 destination port for the input data. Port2 16 is designed as an 8 bit input

1 port 32 and output port 34 with a 3 bit length for control data 36. Similarly,
2 as shown, port3 18 is also an 8 bit input port 38 and output port 40 with a 3
3 bit length for control data 42. Although the length of the data bit and
4 control bit may vary in conventional crossbars, the ports of the crossbars are
5 all generally set up to have identical input and output port configurations
6 across all the ports.

7 [0017] An embodiment of a crossbar designed in accordance with
8 the principles of the present invention is shown in FIG. 2, and indicated
9 generally at 50. For clarification, only 4 ports are shown for the crossbar,
10 specifically port0 52, port1 54, port2 56 and portN 58. As indicated by
11 portN 58, the crossbar can be implemented with as many ports as desired,
12 and the only limitation is the hardware of the computing system. Thus,
13 other implementations of any number of ports for the present crossbar are
14 contemplated and are within the scope of the present invention.

15 [0018] As shown, unlike the prior art, each port of the present
16 crossbar can be customized with different port configurations. In particular,
17 input port0 60 and the output port0 62 of port0 52 are customized to receive
18 "A" data bits and transmit "B" data bits, which indicates different bit
19 lengths for the input port and the output port. Control data 64 generally
20 sent together with data received on the input port0 60 will have "C1" bits.
21 The present invention also includes crossbar control data 66 having "C2"
22 bits that indicates crossbar control information for transferring data from the
23 input port0 62 to an output port0 62 having different port configurations.

24 [0019] The crossbar control data is internal data from the
25 crossbar, which is generated at the time of the design of the crossbar.
26 Unlike the control data 64, the crossbar control data 66 is not sent from the

1 source agents. Rather, the computer engineer provides the crossbar control
2 data at the time when the crossbar with customized ports is created. Thus,
3 the crossbar control data 66 is preferably implemented with the crossbar. In
4 order for the ports with different configurations to work in unison, the
5 crossbar control data 66 is placed internally so that the received data can be
6 correctly formatted for transmission to a port with different configurations
7 (e.g., bandwidth). As a result, even a single port can be customized as an
8 input port and an output port with different port configurations.

9 [0020] The customization and variations of the port
10 configurations can be extended throughout other ports in the crossbar. As
11 shown, port1 54, port2 56 and portN 58 all contain different width
12 configurations. More specifically, port1 54 is customized as a "D" bit
13 width input port 68 and a "E" bit width output port 70 with a "F1" bit width
14 for control data 72 and a "F2" bit width for crossbar control data 74.
15 Similarly, port2 56 is customized as a "G" bit width input port 76 and a "H"
16 bit width output port 78 with a "I1" bit width for control data 80 and a "I2"
17 bit width for crossbar control data 82, and portN 58 is also customized as a
18 "J" bit width input port 84 and a "K" bit width output port 86 with a "L1"
19 bit width for control data 88 and a "L2" bit width for crossbar control data
20 90.

21 [0021] Although all the ports are shown to have different
22 bandwidth configurations, this is an extreme example to show the flexibility
23 of possible implementations for the present invention. However, in
24 practice, the ports will likely have far less variation, because the
25 implementation may be too complex for the needs of most current
26 processing systems. In addition, in this example, the crossbar control data

1 have been shown with different bandwidth data lengths, and they are
2 specific to the ports. Again, the example provided illustrates the many
3 possibilities and flexibility of the present invention. In practice, there is
4 likely one set of crossbar control data for the whole crossbar, containing
5 only a simple control algorithm.

6 [0022] It is also important to note that although width of the ports
7 is shown as the main port configuration that is customized, other port
8 configurations are contemplated as long as the crossbar control data can be
9 processed in the crossbar. For example, port configuration of frequency
10 may be implemented. Because of the hardware limitation of current chips,
11 width of the ports and bandwidth are generally the main configurations for
12 ports in a crossbar. However, as technology expands in the area,
13 customization of other port configurations may be more practical, and these
14 various implementations are within the scope of the present invention.

15 [0023] An overall view of the processing system implemented
16 with the present crossbar is shown in FIG. 3, and indicated generally at 100.
17 The preferred implementation includes one or more virtual channels (two of
18 which are shown) for each port for receiving data from a source agent (not
19 shown). In particular, input port0 60 includes virtual channels 102, 104,
20 input port1 68 includes virtual channels 106, 108, input port2 76 includes
21 virtual channels 110, 112 and input port3 84 includes virtual channels 114,
22 116. As shown, each virtual channel 102, 104, 106, 108, 110, 112, 114, 116
23 is implemented with a First-In-First-Out (FIFO) buffer respective
24 prioritizing devices 118, 120, 122, 124, 126, 128, 130, 132, respective
25 registers 134, 136, 138, 140, 142, 144, 146, 148, and respective
26 multiplexors ("MUXs") 150, 152, 154, 156, 158, 160, 162, 164.

1 **[0024]** The use of virtual channels, MUXs and registers is well
2 established in the art of crossbars. Each input port is divided into multiple
3 virtual channels for receiving data from multiple source agents. Although
4 two virtual channels are shown in FIG. 3 for simplicity, it is typical for each
5 port to be implemented with 5 or more virtual channels. Because each port
6 with the assigned virtual channels is customized differently, a separate
7 register for storing the data in a temporary memory and a MUX for
8 switching configuration internal to the crossbar are preferably implemented
9 for each virtual channel. Depending on the differences and design of the
10 port configurations in the crossbar, an implementation with a fewer number
11 of registers and MUXs may be possible and will be apparent to one skilled
12 in the art.

13 **[0025]** Data received from the virtual channels are then
14 transmitted to a requested destination output port. As shown, there are also
15 separate MUXs 166, 168, 170, 172 that correspond to the output ports 62,
16 70, 78, 86, respectively, which are followed by a 4 to 1 MUX 174. The 4 to
17 1 MUX 174 then transmits the data to a register 176 of one of the
18 destination output ports. Although only one output port is shown for
19 simplicity (i.e., port 170), there would be 4 output ports having 4 separate
20 registers (one shown) that transmit the data to a destination output port,
21 since the crossbar has 4 ports in the example shown in FIG. 2. As shown on
22 each MUX 166, 168, 170, 172, the bandwidth for each output port (i.e., B-
23 bit, E-bit, H-bit, K-bit) is different from the input ports (i.e., A-bit, D-bit, G-
24 bit, J-bit). As a result, in order for data from the input port to be transmitted
25 to the output port having a different port configuration, the data must be
26 first processed using the control information from the crossbar control data.

1 **[0026]** Several implementations are contemplated for the
2 processing of data. The crossbar control data can be stored and processed at
3 several places in this example. The crossbar control data can be processed
4 by either the MUX 166, 168, 170, 172 or the 4 to 1 MUX 174 on the output
5 ports. Because of the configuration of the overall system shown in FIG. 3,
6 the crossbar control data are preferably implemented at the output ports 62,
7 70, 78, 86. However, other implementations at the input ports 60, 68, 76,
8 84 are also contemplated. For example, the crossbar control data can be
9 processed by either the registers (i.e., a shift register) 134, 136, 138, 140,
10 142, 144, 146, 148 or the MUXs 150, 152, 154, 156, 160, 162, 164 of the
11 virtual channels 102, 104, 106, 108, 110, 112, 114, 116. As noted, various
12 alternative implementations are contemplated, and they are within the scope
13 of the present invention.

14 **[0027]** Turning to an important aspect of the illustrated
15 embodiment of the present invention, a flow chart of the preferred
16 functionality of the illustrated embodiment of the present invention is
17 shown in FIG. 3, and indicated generally at 180. After receiving data with
18 the corresponding control data on an input port from a source agent (block
19 182), the control data of the received data are then read (block 184) and
20 determined whether the control data contain valid information for the data
21 received on the input port (block 186). If not, the process is aborted (block
22 188), since the control data contained invalid information. If, however, the
23 information from the control data has been verified as valid (block 186), the
24 requested destination output port for the received data is obtained from the
25 control data (block 188).

1 **[0028]** After the destination output port for the received data is
2 obtained from the control data (block 188), it is next determined whether
3 the input port that received the data has the same port configuration as the
4 obtained destination output port (block 190). If the input port has the same
5 configuration as the output port (block 190), the received data will be
6 transmitted to the destination output port (block 192). If, on the other hand,
7 the input port does not have the same configuration as the output port
8 (block 190), control information is then obtained from the crossbar control
9 data (block 194) for processing the received data to the configuration of the
10 output port (block 196). After the data is processed, the processed data is
11 then transmitted to the output port (block 198).

12 **[0029]** A flow chart illustrating the preferred functionality of the
13 processing step (block 196) shown in FIG. 4 will be explained in FIG. 5. It
14 should be noted that the processing method shown is for processing ports
15 with different width port configurations. However, the present invention
16 contemplates use with other port configurations, and other implementations
17 and methods relating to these port configurations are within the scope of the
18 present invention. Based on port configuration of different widths, the first
19 step in the processing method (block 196) is to determine whether the width
20 of the input port is greater than the width of the output port (block 200). If
21 not, the data are submitted as processed data (block 202) back to the method
22 shown in FIG. 4, since it is not necessary to format data with smaller length
23 bit that can be read by width with greater length bit. In other words, a 24
24 bit width port can read any data bits that are equal to or fewer than 24 bit
25 length, but not data that are more than 24 bit length.

1 **[0030]** However, if the width of the input port is more than the
2 width of the output port (block 200), modification to the data is necessary in
3 order to transmit the data to a port with less available width. Thus, the
4 width of the output port must be ascertained (block 204) in order to format
5 the data from the input port to data configured for the width of the output
6 port (block 206). The formatted data will be submitted as the processed
7 data. In practice, the data will be transmitted by shifting the data bits to the
8 width of the output port. For example, if a 24 bit width input port requests
9 to transmit data to an output port with an 8 bit width, the data will then be
10 shifted or divided into 8 bit for each transmission. Since there are 24 bits
11 total, the data will be transmitted 8 bits at a time until all 24 bits are sent.
12 This can be done either through the use of a MUX or a shift register.
13 Various methods can be used and they will be apparent to one skilled in the
14 art. However, it should be noted that these various other methods are
15 within the scope of the present invention.

16 **[0031]** From the foregoing description, it should be understood
17 that an improved system and method for providing connections between a
18 plurality of customized ports and a plurality of system agents have been
19 shown and described, which have many desirable attributes and advantages.
20 The system and method provide a way for ports in a crossbar to be
21 customized and defined according to the needs and implementations
22 selected by the engineers, creating a more flexible crossbar that is better
23 suited to meet the demands of the various utilization needs of the systems.

24 **[0032]** While various embodiments of the present invention have
25 been shown and described, it should be understood that other modifications,
26 substitutions and alternatives are apparent to one of ordinary skill in the art.

4 **[0033]** Various features of the invention are set forth in the
5 appended claims.